No. 5,629,594, which is a continuation-in-part of U.S. patent application No. 08/257,070, filed June 9, 1994, now U.S. Patent 5,459,382, which is a divisional application of U.S. Application No. 07/984,324, filed December 2, 1992, now U.S. Patent No. 5,389,865.

Please replace the paragraph starting on page 3, line 24 with the following:

Further, by reacting to remote forces present on a slave device, the prior art devices lack the capability of creating a three-dimensional tactile virtual reality environment whereby a user's actions and reactions are related to a simulated world such as simulation of driving or flying functions, simulation of molecular force interactions, or simulations of surgical procedures. U.S. Patent No. 5,044,956 to Behensky et al. discloses a system whereby a steering wheel is used to input positions to a simulation which in turns actuates the wheel in response to simulated artifacts. This system does not disclose or anticipate the simulation and coordination of the six-degrees of freedom required for the arbitrary positioning and orientation of solid objects. Similarly, prior art devices which simulate virtual reality by visual feedback to a user are not capable of accepting tactile inputs and providing tactile force feedback.

Please replace the paragraph starting on page 4, line 12 with the following:

The present invention solves the problems of the prior art by providing a method and system for providing a tactile virtual reality in response to user position and orientation. The present invention further provides a universal device whose kinematics do not necessarily replicate any particular device it might control or simulate. A computer mediated control system is provided which transforms forces, torques, displacements, velocities, and accelerations measured by a simulated environment and applies them to the hand controller or vice. The present invention can effect and control the superposition of translational displacement with force application and angular displacement with torque, thus providing arbitrary, programmed application of forces, torques, and displacements to the user in any direction. This allows the device to be controlled by, and to control, external simulations or models as well as physical remote devices. The invention can also locally simulate virtual force fields generated from

interaction with virtual surfaces and/or boundaries, can provide software programmed position, velocity, force, and acceleration limit stops, and can dynamically shift, rotate, or scale these virtual objects.

Please replace the paragraph starting on page 6, line 24 with the following:

FIGURE 6a presents a top view of the X-Portion of the X-Y table of an embodiment of the manipulator of the present invention;

Please replace the paragraph starting on page 7, line 27 with the following:

FIGURE 11b presents a top view of the roll-stage of the manipulator of an embodiment of the present invention;

Please replace the paragraph starting on page 13, line 14 with the following:

Referring to Figure 10, the pitch stage is shown. Figure 10a presents a front view of the pitch stage and Figure 10b present side view of the pitch stage. The pitch stage is comprised of the pitch motor 160, which is coupled to the pitch gearbox 162 affixed to the yaw-pitch bracket 150. The pitch gearbox includes a pitch spur gear 166 coupled to the pitch motor pinion 168. The output shaft of the gearbox is affixed normal to the vertical arm of the pitch-roll gimbal bracket 170. The weight of the roll axis and the pitch-roll gimbal is compensated by using a constant force spring 172 with a spring spool 174. This does not provide perfect balance accept at the equilibrium position. However, the small centering force is easily overpowered by the pitch motor gear train and holding friction.

Please replace the paragraph starting on page 20, line 9 with the following:

If the interrupt routine determines that it is time to run the servo code, it first checks (in the overrun logic) to see if a previous call to the servo routines is still being processed (this is done via interlocking flags). If the last loop has not yet completed, i.e. there are to many commands or controls to be executed in the user programmed interrupt call-back period, an overrun is signaled and the new interrupt is